OIPE
SEP 26 2002
JC26
PATENT & TRADEMARK OFFICE

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

EX PARTE Hickman

Application for Patent

Filed February 28, 1997

Serial No. 08/810,620

RECEIVED
SEP 27 2002
Technology Center 2100

FOR: METHOD AND APPARATUS FOR COMPUTING OVER A WIDE AREA NETWORK

APPEAL BRIEF

09/27/2002 AWONDAF1 00000011 502207 08810620
01 FC:220 160.00 CH

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail, postage prepaid, in an envelope addressed to: Commissioner for Patents, Washington, D.C. 20231, on September 23, 2002

Signature: Valerie Peterson

Valerie Peterson

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST

The parties in interest is G&H Nevada-Tek, the assignee of record.

II. RELATED APPEALS AND INTERFERENCES

This appeal is related an appeal of a rejection of the claims of USSN 08/798,704, entitled "Method and Apparatus for Computing within a Wide Area Network," also assigned to G&H Nevada Tek. Applicant recommends that the appeals be considered concurrently.

III. STATUS OF THE CLAIMS

Claims 1-18 and 21-22 are pending in this application. All claims have been rejected by the Examiner and are the subject of this Appeal.

IV. STATUS OF THE AMENDMENTS

Applicants did not file any amendments after final rejection.

V. SUMMARY OF THE INVENTION

The claimed invention permits virtually the entire functionality of a computer system to be made accessible to a client computer via a wide area network such as the Internet. (*see*, page 5, lines 7-8) More particularly, the invention permits a computer system to be run as a "virtual machine" through a network browser such as a Netscape® or Internet Explorer® network browser. (*see*, page 6, lines 3-4)

The apparatus of the present includes a number of network accessible computers coupled to a WAN such as the Internet. The functionality of the network available computers can be controlled by a user at a client computer coupled to the network, such that a network available computer becomes a "virtual machine" for the user. A cluster administration computer of the present invention is coupled to the network to monitor the operation of the network available computers. In certain embodiments, the cluster administration computer can reset a malfunctioning network available computer, can match a user's computational needs to that of a suitable network available computer, and/or can provide lists or menus of network available computers. (*see*, for example Fig. 13 and page 24, line 27+)

Great computational and storage efficiencies can be obtained with the network accessible computers of the present invention. For example, since a typical stand-alone personal computer is only used a few hours of the day, by having a number of network accessible computers it is possible to reduce the total number of computers required to service the many individual users. For example, computers that would normally be idle in one time zone can be used by users in another time zone. Furthermore, any user with access to a computer having a network browser would be able to control a powerful network computer from any location having Internet access. Furthermore, the client can choose between many types of computers, such as Windows®, Macintosh®, and Linux® computers, depending upon their needs. The cluster administration computer coordinates and implements these various uses of the network accessible computers of the cluster computer system of the present invention, and allows the sharing of resources. (*see*, for example Fig. 13 and page 25, lines 3-14)

By practicing the claimed system and method, a client computer may control virtually the entire functionality of the network accessible computer with its input device as if the input device were connected to the network accessible computer. Furthermore, the client computer may display images on the monitor within a browser window as if the monitor were connected to the network accessible computer computer. As a result, the network accessible computer becomes a multi-purpose, virtual machine of a user of the client computer.

The present invention provides many advantages. For one, a low cost "thin client" or network computer can be used to control a much more powerful network accessible computer over the Internet. For another, no special software needs to be loaded onto the client computer by the user, since the client computer takes control of the network accessible computer through a network browser interface. Still further, users can share a network accessible computer over the Internet, at substantial cost savings, and users can store their data and documents on the network accessible computer or elsewhere on the Internet, meaning that they can access their "virtual machine" anywhere that they have Internet access.

Since the network accessible computers operate as virtual machines, i.e. appearing to be personal computers right at the user's desktop, a desirable embodiment of the present invention saves the "personal state" of the virtual machines, such that when a user once again logs on to a network accessible computer, it can appear to be the same computer and in the same state as when the user had previously logged onto a network available system. In this

way, the experience of using a network accessible computer is virtually indistinguishable from using a personal computer, from the user's viewpoint. (*see*, for example Figs. 14 and 15, page 26 line 1 to page 27, line 7)

The cluster administration computer facilitates the efficient use of the network accessible computers, and allows a user to identify and connect to a network accessible computer with compatible hardware and software requirements. This can be accomplished automatically, or by presenting the user with a list or menu of options which might meet their needs. The cluster administration computer can further perform resets of malfunctioning network accessible computers. (*see*, for example, page 25, line 29-36)

VI. ISSUES

The issues presented in this appeal are whether the rejections of the claims as set forth by the Examiner are proper. The two issues therefore are as follows:

A. Are claims 1-14, 18, 21 and 22 properly rejected under 35 U.S.C. 103(a) as being unpatentable over Adams and further in view of Frese and Pitkin?

B. Are claims 15-17 properly rejected under 35 U.S.C. 103(a) as being unpatentable over Adams and further in view of Frese, Pitkin and Clark?

VII. GROUPING OF THE CLAIMS

With respect to the first rejection (issue A, above), Applicant proposes the following groups to stand or fall together:

Group 1	Claims 1-7, 9, 12 and 21.
Group 2	Claim 8
Group 3	Claims 10 and 11
Group 4	Claim 13 and 18
Group 5	Claim 14
Group 6	Claim 22

With respect to the second rejection (issue B, above), Applicant proposes the following groups to stand or fall together:

Group 1 Claim 15

Group 2 Claims 16 and 17

VIII. THE CITED ART

A. Adams et al. (U.S. Patent No. 5,913,920)

Adams et al. ("Adams") teaches two computer workstations that are connected together by a communications link, and operates in a fashion very similar to that of the admitted prior art, such as Timbuktu and Carbon Copy (*see* p. 4, lines 6-17 of Applicant's specification). A local workstation includes a window which is used to display a copy of what is currently being displayed on the screen of a remote workstation. Each time an update is made to the screen of the remote workstation, it must be transmitted to the local workstation. A bounding rectangle for the area of the screen changed by the update is determined. If the bounding rectangle is greater than a predetermined size, then a first packet is transmitted from the remote workstation to the local workstation, prior to the transmission of the actual update itself. This is to provide quicker feed-back to users of the local workstation that update data is on its way. This first packet indicates the bounding rectangle for the updated area of screen. The local workstation responds to the first packet by shading the corresponding region of the window which contains the copy of the remote screen, thereby indicating to a user of the local screen that an update is imminent.

B. Frese, II et al. (U.S. Patent No. 5,909,545)

In Frese, II et al. ("Frese") a system and method is disclosed for remotely controlling an *evaluation application program* over a network. The system includes an application interception module and remote display module. The remote display module is transported across the network and executed on the user system in response to a user's request to provide on-demand remote control of an application program. The application interception module captures an I/O stream generated by an application program, converts it to remote control protocol messages and transports them across a network to the remote display module executing in the user system. The remote display module converts the remote control

protocol messages to system calls compatible with the operating system environment for the users computer. Likewise, the remote display module converts system calls to the local resource interface in the user's computer to remote control protocol messages which are transported across the network to the application interception module. The application interception module interface converts the remote control protocol messages to system calls for the application program. In this manner, output from the application program is provided to the user's computer and input actions at the user's computer are provided to the application program. Preferably, the remote display modules and application programs are presented through HTTP servers over a network to a user's system which uses a browser having a JAVA interpreter to execute the remote display module and convert the remote control protocol messages.

Notably, Frese does not allow general purpose access to the computer upon which the evaluation application program resides. For example, a user could not perform a soft reset of the computer system, delete or move files, launch an arbitrary application program, or otherwise perform general purpose functions on the remote computer.

C. Pritkin et al. (U.S. Patent No. 5,341,477)

In Pritkin et al. ("Pritkin") a broker mechanism allocates a plurality of servers, each having an available resource capacity, to a plurality of clients for delivering one of several services to the clients. The broker operates by monitoring a subset of all available servers capable of delivering the requested service. The allocation is based on developing a network policy for the plurality of servers by collecting a local policy for each of the servers. The broker receives client requests for the services and based on the network policy and available resource capacity suggests one of the servers, monitors in its subset for that particular service, to one of the clients making a request. The server suggested enforces its local policy by not allowing any connections exceeding its available resource capacity.

D. Clark et al. (DAWGS – A Distributed Compute Server Utilizing Idle Workstations)

Clark et al. ("Clark") is an article entitled "DAWGS – A Distributed Compute Server Utilizing Idle Workstations", published in The Fifth Memory Computing Conference, April 8-12, 1999, held in Charleston, South Carolina by the IEEE Computer Society Press. The article describes a collection of workstations interconnected by a local area network which

can be utilized as compute servers when left idle by their owners. The system allows users to submit "jobs" for execution on an idle workstation on the LAN by using a distributed scheduler and bidding process to determine on which machine a process should be run. As part of the process, the execution of a job is "checkpointed," so that the job can be moved to another workstation for completion, and is tolerant of machine failure. Clark therefore basically teaches a system for fault-tolerant distributed computing.

IX. ARGUMENTS

A. Claims 1-14, 18, 21 and 22 were improperly rejected under 35 U.S.C. 103(a) as being unpatentable over Adams and further in view of Frese and Pitkin.

This rejection will be considered on a group-by-group basis. As discussed below, Applicant believes that all pending claims are patentable over Adams in view of Frese and Pitkin.

1. Group 1 Claims 1-7, 9, 12 and 21

Group 1 includes claims 1-7, 9, 12 and 21 of which 1 is the independent claim. This group therefore stands or falls with respect to claim 1.

With reference to claim 1, it is clear that the claim includes two major elements: a plurality of network accessible computers each of which can have its functionality controlled by a client computer and a cluster administration computer monitoring the operation of the network accessible computers. Each of the network accessible computers has its own unique address with respect to the network. The client computer controls the functionality of a network accessible computer after being downloaded a client program. Inputs from the client computer generate inputs on the network accessible computer, and image information of the network accessible computer is viewed in a window of the client computer, such that the client computer takes over the general functionality of the network accessible computer to run general purpose application programs and to control the operating system of the network accessible computer.

Adams teaches the remote control of a computer over a token ring network. That is, Adams teaches a very different type of system and, in fact, teaches away from the invention claimed by Applicant. With Adams, an application on a remote

workstation can be controlled by a local workstation to provide a "collaborative work environment." The focus of Adams' invention is that when doing so, the lag time between creating an event at the local workstation (e.g. a mouse click) and receiving the results of that event from the remote workstation (e.g. an update to the display) can cause the operator of the local station to repeat the event unnecessarily.

Adams discusses the World Wide Web (WWW) which is, of course, supported by Internet protocols. More particularly, Adams asserts that the approach of the WWW "is fundamentally driven by the client (i.e. receiving) terminal ... [and] ... is not directly applicable to a collaborative working environment." Col. 1, lines 62-65. That is, Adams does not apply his collaborative working environment processes to an Internet or similar wide area network environment and, in fact, teaches away from such an environment, since to do so would worsen the very problem he was trying to solve.

It is therefore very clear that Adams lacks many of the essential limitations as recited in claim 1. First, Adams does not teach a plurality of network accessible computers, each of which has a unique address, that can be accessed by a client computer. In fact, Adams teaches the opposite: that a single remote computer workstation can simultaneously be controlled by and update screen images of multiple local computer workstations.

> "Note also that the invention is not limited to systems having only two workstations, but is equally applicable to systems where the original update is copied to multiple other screens. In such cases, the indication of update area will generally be provided to all the multiple other screens, although possibly if the links to some terminals are much slower than the links to other terminals, it might be sensible for only those terminals connected by slow links to receive the update area message prior to the actual update itself." (Adams, column 4, line 66 – column 5, line 8)

Second, Adams does not hint or suggest the use of a cluster administration computer. As such, the invention of Adams is only a slight variation over the admitted prior art of Timbuktu and Carbon Copy (*see*, page 4, lines 6-17 of Applicant's specification). Adams is clearly a "point-to-point" system, as disclosed by Applicant with respect to the prior art.

Frese does not cure the deficiencies of Adams as prior art to the claims under appeal. First, it is noted that Frese has a priority date of January 19, 1996, while the

present invention claims dual priorities of February 16, 1996 and March 6, 1996. Applicant therefore reserves the right to swear behind Frese, but is firmly convinced that there is no need to do so, as explained below.

Frese permits a user (at a client computer) to test and evaluate customized application programs. The user is not given full control over the computer hosting the application program, and the application programs provided for evaluation purposes were specifically designed for remote execution. Since Frese is directed to controlling an evaluation copy of an application program, it is not designed control the functionality of a host computer, but merely to operate a specifically designed evaluation application program on a remote computer. Furthermore, Frese, like Adams, does not teach a plurality of network accessible computers, nor a cluster administration computer to monitor the plurality of network accessible computer. Therefore, Frese, like Adams, does not include many essential limitations of the invention claimed by Applicant.

Frese does teach the use of Internet protocols for transmitting between a service publisher server (which hosts a specifically designed evaluation application program) and a user computer. Upon a request from the user computer, a remote application server is started on the service publisher server, along with an application program to be evaluated, and an application intercept module. A remote display module is sent to the user computer to allow communication with the application program being evaluated.

It is improper to combine the teachings of Adams and Frese because they mutually teach away from each other. That is, Adams specifically teaches away from utilizing Internet protocols, browsers and Applets to permit a local computer to take over the control of a remote computer, and Frese specifically discloses a methodology which only allows limited control of a specified application program on a remote computer, rather than general control over the remote computer itself. To do so would defeat the purpose of Frese, which is to allow a user to evaluate a special evaluation application program and then purchase the program to run on his own computer. If Frese allowed the general functionality of the remote computer itself to be accessed, the user would have no need to buy the application software, defeating the purpose of his invention, and obviating the need for the very specialized software that he developed to safely control the evaluation use of specified application programs. Thus, the use of Frese's Internet technology is contrary to the

teachings of Adams, and the use of Adams' remote control technology is contrary to the teachins of Frese. As such, it is improper to combine these two references.

Pitkin also does not cure the deficiencies of Adams and Frese. Pitkin teaches a broker (server) which diverts requested services from an accessing client to an available (servicing) servers. That is, when a service request is received from a client, it is diverted to a server capable of servicing the request. In a preferred embodiment of Pitkin, each server has a specialized skill that it can provide. There is no hint or suggestion that the functionality of the server be taken over by the client and, in fact, there would be no reason to do so since the servers are not general purpose computers but, rather, special purpose servers.

It is, in fact, improper to combine Pitkin with Adams and Frese, since Pitkin specifically teaches that it is contrary to the purpose of his invention for a client computer to be directly associated with a single host computer:

> "One known type of broker operates by assigning an entire server to a client irrespective of the capacity needed by the client. A problem with the above broker method is the inefficient use of network devices." (Pikin, column 2, lines 22-26).

That is, according to the invention of Pitkin, it would be inefficient use of network devices to have a client computer take over and control the functionality of a host computer.

It is therefore very clear that the claims of Group 1 not only include major elements not found in the cited art, e.g. a plurality of network accessible computers each of which can have its functionality controlled by a client computer and a cluster administration computer monitoring the operation of the network accessible computers, but that the cited art cannot even be combined as suggested as they mutually teach away from each other. Applicant therefore asserts that the rejection of the claims of Group 1 was improper, and that the rejection should be withdrawn.

2. Group 2 Claim 8

Group 2 includes claim 8, which is indirectly dependent upon claim 1. As such, the reasons for patentability of claim 8 are at least the same as set forth with respect to the Group 1 claims, above. In addition, there is no hint, suggestion, or even need, for a host machine to be reset by a monitoring administration computer, in the prior art.

Adams does not permit the resetting of a host machine by a monitoring cluster administration computer. That is, there is no cluster administration computer or equivalent to begin with. If the host machine of Adams "locks up", there is no way for the user to reset the machine. This would require a hardware reboot at the host machine.

Frese would not allow any type of resetting by a client computer, since the user is only communicating in a limited fashion with an evaluation application program, and does not have access to system functions such as reset. To allow this functionality in Frese would be dangerous, since all of the monitoring and control programming could then be bypassed. Further, as noted above, Frese does not include a cluster administration computer.

With Pitkin, there is no disclosure that the broker has the ability to reset a server. In fact, since the servers of Pitkin are designed to service multiple users it would be very dangerous to allow the broker this kind of power over the servers. There simply is no analogy between the broker of Pitkin and Applicant's cluster administration computer.

It is, in fact, very illuminating that none of the cited art allows for the monitoring and resetting of selected network accessible computers by client computers. This is because none allow network accessible computers to be taken over by client computers as "virtual machines" which appear to function the same way as a computer system on the user's desktop. As such, the only fundamental difference between a computer on the user's desktop and the control of the functionality of a network accessible computer of the present invention is the ability to reset the network accessible computer if it malfunctions or "freezes." Obviously, in the case of a freezing, the user cannot accomplish a reset. The cluster administration computer, which is monitoring the functionality of the network accessible computers, will detect the problem and implement a hardware reset of the computer.

As such, the cited art does not even address the same problem which is solved by the invention of claim 8 and, as such, the rejection of this Group 2 claim was clearly in error and should be withdrawn.

3. Group 3 Claims 10 and 11

Group 3 includes claims 10 and 11, which are indirectly dependent upon claim 1. As such, the reasons for patentability of claims 10 and 11 are at least the same as set forth with respect to the Group 1 claims, above. In addition, there is no hint, suggestion, or even need, for a cluster administration computer to coordinate the sharing of at least one local resource, such as a data storage device.

Neither Adams nor Frese teaches a cluster administration computer nor a plurality of network accessible computers and, therefore, cannot teach that a cluster administration computer controls a local resource for the network accessible computers. With respect to Pitkin, there is no teaching that the broker controls a local resource for the servers, nor that the servers share any resources. Pitkin, in fact, suggests that the resources of the servers are independent in order to enhance reliability. That is, Pitkin teaches that two or more servers can be suggested to a client such that if one server fails, the other can be used.

> "Failure of a server can occur due to ... failure of the hardware, line damage, etc. ... The includes of a second server suggestion provides the client with an alternate server without having to recontact the broker." (Pitkin, col. 11, lines 5-12)

It is therefore abundantly clear that the cited art does not show, suggest, or even remotely address the invention of claims 10 and 11, and therefore should be withdrawn.

4. Group 4 Claims 13 and 18

Claim 13 is directed to a method of the present invention for providing access to host computers over a network. Claim 18 is the corresponding computer program product. As such, the Group 4 claims stand or fall with method claim 13. Since claim 13 is a method counterpart to the system claims of Group 1, many of the arguments made with respect to Group 1 are incorporated herein and will not be repeated.

Claim 13 specifies that a request from a client for a host computer to serve as a "virtual machine" is received by a cluster administration computer. That is, the relationship of the host computer to the client computer is such that, after the client computer is downloaded a client program, the input devices of the client computer generate inputs to the host computer, and image information generated by the host computer can be viewed by the client computer. The cluster administration computer then determines a suitable host

computer, and informs the client computer of the network address of the suitable host computer. The cluster computer further monitors the functionality of the plurality of network accessible computers, which can serve as the host computers.

As described above in greater detail with respect to Group 1, Adams teaches the remote control of a computer over a token ring network. That is, Adams teaches a very different type of system and, in fact, teaches away from the invention claimed by Applicant. It is therefore very clear that Adams lacks many of the essential limitations as recited in claim 13. First, Adams does not teach receiving of a request for a host computer by a cluster administration computer, nor does it teach the downloading from the host computer to the client computer a client program which allows the client computer to take over the functionality of the host computer. Adams further does not teach the determination of a suitable host computer, nor informing the client computer concerning the suitability of a host computer. Adams also does not address the monitoring of a number of network accessible computers by a cluster administration computer.

Frese does not cure the deficiencies of Adams as prior art to the claims under appeal. First, it is noted that Frese has a priority date of January 19, 1996, while the present invention claims dual priorities of February 16, 1996 and March 6, 1996. Applicant therefore reserves the right to swear behind Frese, but is firmly convinced that there is no need to do so, as explained below.

Frese permits a user (at a client computer) to test and evaluate customized application programs. The user is not given full control over the computer hosting the application program, and the application programs provided for evaluation purposes were specifically designed for remote execution. For example, a user cannot run application programs of their own choice, nor provide operating system commands such as providing a soft reboot of the system, moving or deleting files, etc. That is, since Frese is directed to controlling an evaluation copy of an application program, it is not designed control the functionality of a host computer but, merely to operate a specifically designed evaluation application program on a remote computer. Furthermore, Frese, like Adams, does not teachr a cluster administration computer to monitor a plurality of network accessible computer. Therefore, Frese, like Adams, does not include many essential limitations of the invention claimed by Applicant.

Frese does teach the use of Internet protocols for transmitting between a service publisher server (which hosts a specifically designed evaluation application program) and a user computer. Upon a request from the user computer, a remote application server is started on the service publisher server, along with an application program to be evaluated, and an application intercept module. A remote display module is sent to the user computer to allow communication with the application program being evaluated.

As noted above, it is improper to combine the teachings of Adams and Frese because they mutually teach away from each other. That is, Adams specifically teaches away from utilizing Internet protocols, browsers and Applets to permit a local computer to take over the control of a remote computer (see above), and Frese specifically discloses a methodology which only allows limited control of a specified application program on a remote computer, rather than general control over the remote computer itself. That is, the Internet processes of Frese are specifically said in Adams to be disadvantageous, and the control functionality of Adams would defeat the purpose of Frese to allow limited evaluation of specially adapted application programs.

Pitkin also does not cure the deficiencies of Adams and Frese. Pitkin teaches a broker (server) which diverts requested services from an accessing client to an available (servicing) servers. That is, when a service request is received from a client, it is diverted to a server capable of servicing the request. In a preferred embodiment of Pitkin, each server has a specialized skill that it can provide. There is no hint or suggestion that the functionality of the server be taken over by the client and, in fact, there would be no reason to do so since the servers are not general purpose computers but, rather, special purpose servers. As such, the servers are meant to service many users simultaneously and, in no event, would a single user be allowed to take over the full functionality of the server.

It is therefore clear that the rejection of the Group 4 claims over Adams, Frese and Pitkin was in error, and should be withdrawn.

5. Group 5 Claim 14

Group 2 includes claim 14, which is directly dependent upon claim 13. As such, the reasons for patentability of claim 14 are at least the same as set forth with respect to the Group 4 claims, above. In addition, there is no hint, suggestion, or even need, for a process

wherein a client computer provides desired requirements which are compared to the characteristics of host computers on a computer network.

With respect to Adams, a user connects to a host, suitable or not. With Frese, the user is testing an evaluation application program and, as such, the concern with suitability with the host machine is moot. With Pitkin the user does not specify requirements at all. The broker of Pitkin merely suggests one or more available servers that can provide a desired type of service.

Again, the disclosures of Adams, Frese and Pitkin are not addressing the same type of problem as that solved by claim 14. That is, since a client machine takes over the functionality of a network accessible computer, it should be determined that the network accessible computer has the right hardware and software to support the user. For example, if the client wishes to run software designed for an Apple® Macintosh® computer, the client should specify that desire and the cluster administration machine should select a network accessible computer that is hardware, operating system, and application program compatible with the Apple Macintosh computer. Nothing in Adams, Frese, or Pitkin would support or suggest this functionality.

Applicant therefore respectfully submits that the rejection of claim 14 was also in error, and should be withdrawn.

6. Group 6 Claim 22

Group 5 includes claim 22, which is indirectly dependent upon claim 1. As such, the reasons for patentability of claim 1 are at least the same as set forth with respect to the Group 1 claims, above. In addition, there is no hint, suggestion, or even need, for a process wherein a cluster administration computer is operative to create a list of network accessible computers. First, as set forth above, none of the cited art teaches a cluster administration computer for a plurality of network accessible computers. Further, there is no teaching in the cited art for a user to go to a cluster administration computer in order to obtain a list of available network accessible computer which can serve as "virtual machines" for the user. The closest analogy is the "broker" of Pitkin which designates one or more servers that can perform a certain service for a user, but this is far different from the creation of a list of

network accessible computers from which a user can choose a desired network accessible computer to serve as a host virtual machine.

It is therefore clear that the rejection of claim 22 is also in error, and should be withdrawn.

B. Claims 15-17 were improperly rejected under 35 U.S.C. 103(a) as being unpatentable over Adams and further in view of Frese, Pitkin and Clark.

This rejection will also be considered on a group-by-group basis. As discussed below, Applicant believes that all pending claims are patentable over Adams in view of Frese, Pitkin and Clark.

1. Group 7 Claim 15

Claim 15 adds the limitation of the "personal state" to the method of claim 14. Claim 15 is therefore patentable over the cited art for at least the same reasons as set forth with respect to Group 5 of the first rejection (Issue A), as set forth above.

Clark was cited by the Examiner as saving a "personal state." Applicant strongly traverses this assertion. Clark in no way saves a personal state of a client's "virtual machine," i.e. the personal state of a user who is remotely controlling a network accessible computer's functionality. With Applicant's invention, the state of a "virtual machine" is saved such that when the user, once again, takes over the functionality of a network accessible computer, it will be in essentially the same state as when he last used it. Clark's process, in marked contrast, is more like a fail-safe function, wherein a running application program is "checkpointed" (i.e. the operating parameters are saved) periodically during its execution so that it can be restarted after, for example, a power failure. Alternatively, the checkpointed application program can be sent to another host machine for further processing. Clark is basically a distributed computing system wherein an application program is sent from a client machine to one or more host machines with robust fail-safe functionality. Clark in no way teaches the saving of a personal state of a remotely controlled computer system such that a client can repeatedly access a virtual machine as if it were his own personal computer.

Claim 15 is clearly patentable over the combination cited by the Examiner and, as such, its rejection was in error. Applicant respectfully requests that the rejection of claim 15 be withdrawn.

2. Group 8 Claim 16 and 17

Claim 16 adds the limitation of monitoring a plurality of network accessible computers and resetting the computer if it is determined not to be functioning properly. Claim 17 is the computer program counterpart of claim 16. Claim 16 is dependent upon claim 15 and therefore is patentable for at least the same reasons as set forth with respect to Group 7. Furthermore, it is similar to Group 2, claim 8 in that it calls for the resetting of a network accessible computer, and the arguments related to Group 2 are incorporated herein.

Adams does not permit the resetting of a host machine by a monitoring function. If the host machine of Adams "locks up", there is no way for a user to reset the machine with the Adams system.

Frese would not allow any type of resetting by a client computer, since the user is only communicating in a limited fashion with an evaluation application program, and does not have access to system functions such as reset. To allow this functionality in Frese would be dangerous, since all of the monitoring and control programming could then be bypassed.

With Pitkin, there is no disclosure that the broker has the ability to reset a server. In fact, since the servers of Pitkin are designed to service multiple users it would be very dangerous to allow the broker this kind of power over the servers. There simply is no analogy between the functionality of the broker of Pitkin and Applicant's applicant's claimed monitoring and resetting process.

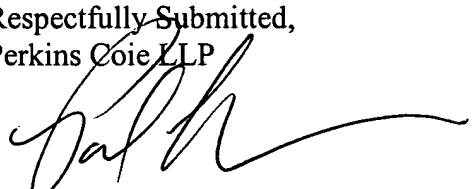
Finally, Clark does not address the problem of monitoring and resetting a host computer. In fact, if a host computer malfunctions, Clark transfers an application program, in mid-execution, to another host computer, which is a primary purpose of his "checkpoint" function.

As such, the cited art does not even address the same problem which is solved by the invention of claim 16 and, as such, the rejection of this Group 8 claim was clearly in error and should be withdrawn.

X. CONCLUSION

As noted, none of the cited art, either alone or in combination, can be said to render obvious the appealed claims. Accordingly, Applicant believes the rejections to be in error, and respectfully requests the Board of Appeals and Interferences to reverse the Examiner's rejections of the claims on appeal.

Respectfully Submitted,
Perkins Coie LLP



Paul L. Hickman
Reg. No. 28,516

Paul L. Hickman
Perkins Coie LLP
Customer No. 22918
101 Jefferson Drive
Menlo Park, California
94025-1114
Direct: 650-838-4443
Fax 650-838-4350

APPENDIX A - THE APPEALED CLAIMS

The claims under appeal (claims 1-18, 21 and 22) are as follows:

1. (amended) A cluster computer system comprising:

a plurality of network accessible computers each having a unique address with respect to a network, each including a central processing unit and non-volatile memory, where each of said network accessible computers is coupled to said network, where said network accessible computers implement host computer programs which permit the network accessible computers to operate as host computers for client computers coupled to said network, where a client computer controls the functionality of a host computer after being downloaded from said host computer a client program to run on said client computer that includes the ability to communicate with said host computer program, whereby an input device of said client computer can be used to generate inputs to said host computer, and such that image information generated by said host computer can be viewed in a window of said client computer; and

a cluster administration computer coupled to said network to monitor the operation of said network accessible computers.

2. A cluster computer system as recited in claim 1 wherein said plurality of network accessible computers are coupled to said network with a corresponding plurality of communication channels.

3. A cluster computer system as recited in claim 1 wherein said plurality of network accessible computers also each include volatile memory and data bus controllers.

4. A cluster computer system as recited in claim 1 wherein said network is a TCP/IP protocol network, and wherein said host computer programs are responsive to keyboards and pointing devices of said client computers as transmitted to said host computers over said TCP/IP protocol network under the control of client programs running on said client computers, said host programs transmitting said image information to said client computers over said TCP/IP protocol network for display in browser windows of browser programs running on said client computers.

5. A cluster computer system as recited in claim 4 wherein said client programs are transmitted to said client computers over said TCP/IP protocol network.

6. A cluster computer system as recited in claim 5 wherein said client programs are Java Applet programs.

7. A cluster computer system as recited in claim 4 wherein said cluster administration computer is operative to control at least one function of said network accessible computers.

8. A cluster computer system as recited in claim 7 wherein said at least one function is to reset a selected network accessible computer.

9. A cluster computer system as recited in claim 4 wherein said cluster administration computer is coupled to said network to receive inputs from other computer systems coupled to said network.

10. A cluster computer system as recited in claim 4 wherein said cluster administration computer serves to coordinate the sharing of at least one local resource by said network accessible computers.

11. A cluster computer system as recited in claim 10 wherein said at least one local source is a data storage device.

12. A cluster computer as recited in claim 4 wherein said cluster administration computer is running a cluster administration program which administers the connection of a client computer to a host computer.

13. (amended) A method for providing access to host computers by client computers over a computer network comprising:

receiving a request for a host computer coupled to a computer network from a client computer coupled to said computer network, said request received by a cluster administration computer, wherein the relationship of said host computer to said client computer is to be such that after said client computer becomes associated with a host computer by being downloaded from said host computer a client program that includes the ability to communicate with a host computer program running on said host computer, an input device of said client computer can be used to generate inputs to said host computer, and such that image information generated by said host computer can be viewed by said client computer;

determining a suitable host computer for said client computer by said cluster administration computer;

informing said client computer of the network address of said suitable host computer by said cluster administration computer, whereby said client computer can become associated with said host computer; and

monitoring the functionality of a plurality of network accessible computers by said cluster administration computer.

14. A method for providing access to host computers by client computers over a computer network as recited in claim 13 wherein determining a suitable host computer includes receiving the desired requirements for a host computer from said client computer, and comparing said desired requirements to the characteristics of available host computers on said computer network.

15. A method for providing access to host computers by client computers over a computer network as recited in claim 14 further comprising loading a personal state of a client using said client computer into said network accessible computer that will serve as said suitable host computer.

16. (amended) A method for providing access to host computers by client computer over a computer network as recited in claim 15 whereby monitoring the functionality of a plurality of network accessible computers includes resetting a network accessible computer if it is determined that it is not functioning properly.

17. (amended) A computer program product comprising a computer readable media having program instructions embodied on said media for implementing the method of claim 16.

18. (amended) A computer program product comprising a computer readable media having program instructions embodied on said media for implementing the method of claim 13.

21. (added) A cluster computer system as recited in claim 5 wherein said cluster administration computer is operative to process a TCP/IP compatible data packet received over said TCP/IP protocol network, where said cluster administration computer is operative to determine the origin and destination of said TCP/IP data packet.

22. (added) A cluster computer system as recited in claim 4 wherein said cluster administration computer is operative to create a list of available network accessible computers.